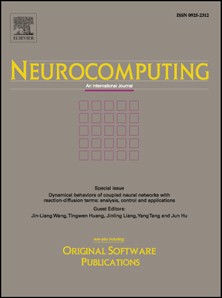
Face Recognition with Dense Supervision Communicated by Dr Zhen Lei

# Journal Pre-proof



Face Recognition with Dense Supervision

Zhiqiang Liu, Weijun Hong, Hongzhou Zhang, Jianhui Ma PII: S0925-2312(19)31761-8

DOI:https://doi.org/10.1016/j.neucom.2019.12.052 Reference: NEUCOM 21681

To appear in: *Neurocomputing*

Received date: 11 September 2019

Revised date: 18 November 2019

Accepted date: 11 December 2019

Please cite this article as: Zhiqiang Liu, Weijun Hong, Hongzhou Zhang, Jianhui Ma, Face Recognition with Dense Supervision, *Neurocomputing* (2019), doi:<https://doi.org/10.1016/j.neucom.2019.12.052>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier B.V.

Input

Component Features

Backbone Network



Feature

Map

𝑓1

𝑓2

𝑓3

𝑓4

𝑓5

𝑓6

𝑓7

𝑓8

𝑓9

Rank by

inconsistency

Sampled with **High** Probability

to backward the loss

Major Group

𝑓1 𝑓3 𝑓7 𝑓8

Minor Group

𝑓2 𝑓4 𝑓5 𝑓6 𝑓9

Sampled with **LOW** Probability

to backward the loss

* We propose to train a family of component features for face recognition within a shared network, by applying dense supervision on top of the output tensor of the backbone network.
* We propose a metric to evaluate the component feature consistency, which is defined as the sum of pair-wise KL divergences between one component feature’s softmax probability and those of the others.
* We use the proposed feature consistency to acquire the importance probability for back-propagating the corresponding loss of one component feature. Correlation between features are reduced thus the component features become more discriminative.
* Experimental results show that our approach outperforms the corresponding single supervision baseline by a large margin, and even performs on par with its multi-patch ensemble counterpart. Compared with state-of-the-art methods which utilize strict supervision, our approach is still competitive, especially in large scale face retrieval and cross-pose face matching.

Face Recognition with Dense Supervision*Y*

ZhiqiangLiu *a,b,c,d*,×,WeijunHong *c*,HongzhouZhang *c*,×× andJianhuiMa *e*

*aGraduate School, People’s Public Security University of China, Beijing 100038, China. bCollege of Computer Science and Technology(Software College), Henan Polytechnic University, Jiaozuo 454003, China. cSchool of Information Technology and Cyber Security, People’s Public Security University of China, Beijing 100038, China. dDepartment of Electronic Engineering, Tsinghua University, Beijing 100084, China. eBureau of Science and Technology Information, Ministry of Public Security of China, Beijing 100741, China.*

A R T I C L E I N F O

*Keywords*:

face recognition dense supervision feature consistency importance sampling

## Introduction

A B S T R A C T

Recent advances in face recognition mostly concentrate on designing more discriminative loss func- tions or adding normalization on features/weights to make a single feature more accurate. In this work, inspired by the frequently used multi-patch ensemble method for face recognition and part-based mod- els for person re-identification, we propose a novel training strategy to enhance the discriminability of deeply learned feature from another perspective, namely learning under dense supervision. The main idea is to apply multiple classification loss on top of multiple component features extracted from a single network. Ideally, each component feature is expected to be accurate and has low correla- tion with the others. To this end, we first design a metric called feature consistency to evaluate the correlation between one component feature and the others, which is defined as the sum of distance be- tween one component feature and the others, where the distance here is measured with KL divergence between the corresponding softmax probabilities. Then we use feature consistency to select which component feature to sample for one learning pass by importance sampling. Dense supervision sig- nificantly outperforms the single supervision baseline and even performs on par with its multi-patch ensemble counterpart which has much more parameters (such as 9x). Our experimental results match state-of-the-art performance on LFW, YTF, MegaFace and outperform the others on LFW BLUFR and VGGFace2 pose protocol, thereby achieving state-of-the-art. Specially, results on VGGFace2 also show the superiority of dense supervision on cross-pose face matching.

the sample is not close enough to other samples of the same

Face recognition is an important research topic in com- puter vision. In the past decades, we have witnessed remark- able improvements in the field of face recognition, owing to the dramatic increase in available training data[1,2,3], ad- vances in network architecture design[4,5] and progress in formulating more discriminative loss functions[6,7,8,9].

Some benchmarks for face recognition, such as LFW[10] for verification and MegaFace[11] for retrieval, have even

reached saturation, with 99*.*83% verification accuracy and

98*.*06% rank-1 accuracy[12] respectively.

To reach such high performance, one of the key problems is the model’s discriminability in unseen samples. Typically, a simple but effective way to learn face representation with barely satisfactory discriminability is to train a deep neu- ral network in supervised classification manner with soft- max cross-entropy loss. However, when we preform face retrieval in large scale database using such face representa- tion, the results turn to be unacceptable. The reason is that softmax loss is not *strict* enough. Once a sample is clas- sified correctly, the loss is prone to be negligible, even if *Y* This document is the results of the research projects funded by the National Key R & D Program of China and the Basic Special Project of the

Ministry of Public Security of China.

×Corresponding author at: College of Computer Science and Tech-

nology(Software College), Henan Polytechnic University, Jiaozuo 454003, China. Department of Electronic Engineering, Tsinghua University, Bei- jing 100084, China.

××Corresponding author.

[2459830187@qq.com](mailto:2459830187@qq.com) (Z. Liu); [zhang\_ppsuc@163.com](mailto:zhang_ppsuc@163.com) (H. Zhang)

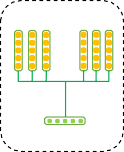
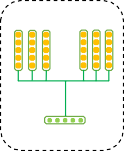
ORCID(s): 0000-0002-1883-2871 (Z. Liu)

class. A feasible solution to address this problem is to for- mulate more strict loss functions that encourage samples in the same class to be close and those in different classes to be far away. Such kind of loss functions have been proved to be effective in promoting the discriminability of the model in several recent works, for instance, triplet loss[6] which op- timizes Euclidean distance and large margin[8]/angular[9] softmax which optimize Cosine distance.

While most of the recent works[6,8,9] concentrate on designing more discriminative loss functions, this paper pro- poses to boost the discriminability of deeply learned fea- ture from a novel perspective, namely, training with dense supervision. The main idea is to divide the output tensor of the backbone network into multiple non-overlapping vec- tors, and each vector is a separated feature for an input face. This is inspired by the frequently used multi-patch ensem- ble method in face recognition[13]. In order to take full ad- vantage of the local information in faces, we usually crop a face into multiple patches and use them to train multiple independent networks. While testing, the output features are fused by directly concatenating, PCA, or learning cer- tain form of transformation. The sizes and locations of the patches remain to be tuned as hyperparameters by cross vali- dation, which is a quite time-consuming procedure. Another drawback of multi-patch ensemble is the increased compu- tational cost, making it unaffordable to deploy large model ensemble in practice. These problems motivate us to develop a method to train a simple but effective ensemble model. To this end, we train multiple features which have a shared back-

Zhiqiang Liu et al.: *Preprint submitted to Elsevier* Page 1 of 12

**Figure 1:** Comparison between multi-patch ensemble and the proposed dense supervision. (a) In multi-patch ensemble, a face is cropped into multiple patches and multiple networks are trained using these patches. Features extracted from different networks are fused to get a final face representation when test- ing. (b) In our proposed dense supervision, the output tensor of the *single* backbone network is split into multiple vectors and we apply multiple supervision on top of them. These vec- tors are extracted as component features and fused to get a final face representation when testing.



PCA

Final feature

CNN

Testing

…

(c) Dense Supervision

CNN

PCA

Final feature

CNN

Testing

…

(b) Multi-patch Ensemble

CNN

Final feature

CNN

Testing

(a) Single Supervision

Supervision

Supervision

Supervision

Supervision

Supervision

Supervision

Supervision

…

…

bone network, by pixel-wisely dividing the output tensor of the backbone network and apply dense supervision on top of each individual vector. We name these vectors component features. Fig.1a) and b) compare the multi-patch ensemble method and our method.

In order to train multiple component features with a shared backbone network, a most straightforward solution is sim- ply to apply classification loss upon each component feature. Such naive strategy brings marginal improvements with re- gard to the single supervision baseline (See Section.4), showing that it is feasible to enhance the learned feature by applying dense supervision. However, to step further in this direction, we find it important to design a smarter strategy to dynamically balance the learning process of each com- ponent feature rather than treating them as equal. Actually, since each component feature corresponds to a facial region, the visibility and the quality of the corresponding region de- termine how useful this component feature is. If such qual- ity, or importance, of component features can be estimated, we could weight its loss with the importance accordingly. In this work, to estimate the importance of each component fea- ture without supervision, we design a metric to compute the inconsistency of each component feature with regard to the others, and use the inconsistency as the importance. Then component features with high importance are sampled to be trained with high probability, and vice versa. This training strategy brings more improvements on dense supervision. Our experimental results match state-of-the-art performance

on LFW [10], YTF [14], MegaFace [11] and outperform the others on LFW BLUFR [15] and VGGFace2 [3] pose proto- col, thereby achieving state-of-the-art. Specially, results on VGGFace2 also show the superiority of dense supervision on cross-pose face matching.

The contributions of this work are summarized as fol- lows.

* We propose to train a family of component features for face recognition within a shared network, by applying dense supervision on top of the output tensor of the backbone network.
* We propose a metric to evaluate the component feature consistency, which is defined as the sum of pair-wise KL divergences between one component feature’s soft- max probability and those of the others.
* We use the proposed feature consistency to acquire the importance probability for back-propagating the cor- responding loss of one component feature. Correla- tion between features are reduced thus the component features become more discriminative.
* Experimental results show that our approach outper- forms the corresponding single supervision baseline by a large margin, and even performs on par with its multi-patch ensemble counterpart. Compared with state- of-the-art methods which utilize strict supervision, our approach is still competitive, especially in large scale face retrieval and cross-pose face matching.

The remainder of this paper is organized as follows. In Section2, we introduce the related works. The proposed ap- proach is presented in Section3. In Section4, we present experimental results and discussions. Finally, a brief con- clusion is drawn in the last section.

## Related work

**Deep face recognition**. Great progress has been made in face recognition in recent years, owning to increasing data[1, 3,2], sophisticated network architectures[4,5,16,17], well- designed normalization techniques [18] and loss functions[19, 6,7,20,9]. Among these, Most researchers concentrate on designing loss functions. [19] proposes to add contrastive loss[21] as verification supervision and jointly train classi- fication and verification. In [6], triplet loss is used instead of softmax loss to learn discriminative face embedding with huge amount of data. [7] proposes center loss to make sam- ples of the same class cluster together, which is usually used together with softmax loss. More recent works tend to mod- ify the vanilla softmax loss by adding more strict objective, such as introducing large angular margin between different classes[8,9]. In this paper, we try to view face recognition from another perspective. Instead of proposing novel loss functions, we study on how to apply dense supervision on one network to make the feature more discriminative.

**Part-based models.** Part-based models are popular in recognition tasks such as face recognition[13,22] and person

Input

Component Features

Backbone Network



Feature

Map

𝑓1

𝑓2

𝑓3

𝑓4

𝑓5

𝑓6

𝑓7

𝑓8

𝑓9

Rank by

inconsistency

Sampled with **High** Probability

to backward the loss

Major Group

𝑓1 𝑓3 𝑓7 𝑓8

Minor Group

𝑓2 𝑓4 𝑓5 𝑓6 𝑓9

Sampled with **LOW** Probability

to backward the loss

**Figure 2:** An overview of the proposed approach. An image is first input to the backbone network and form an output feature map. Then the feature map is point-wisely split into multiple vectors called component features. We then rank the component features by a metric called feature consistency, which is introduced in Sec.3.2. Component features with different ranks are categorized into major group and minor group. In different groups component features are sampled with different probability, and the corresponding losses are back-propagated.

re-identification[23,24,25,26]. We summarize part-based models as a kind of ensemble methods which take different parts of the images as input and train multiple networks with the same topology structure. According to how many param- eters between the networks are shared, we categorize these methods into three types.

The first type is that the models are independently trained and do not share any parameters, as in [13] 60 face patches with different locations and sizes are cropped to train an en- semble of 60 independent networks. This kind of ensemble works well but suffers from large computation cost.

For saving computation there comes the second type, part-based models. In these works, component models share low and middle level features in a trunk network and have in- dependent high level branch networks which are not shared.

[22] proposes such structure for video face recognition. Sim- ilar structures are also adopted in [27] and [23] where the trunk network is used to extract feature maps and features of parts are cropped with ROI pooling then input to branch networks for further feature learning.

The third type, which is the most related to our dense su- pervision, is that models are trained with totally shared net- work. Some works in person ReID have validate the feasibil- ity of such method. For example, in [25] the output tensor of the trunk network is partitioned to 6 regions via a variation of attention mechanism and in each partition the features are pooled to a part feature averagely. In[28], the output ten- sor of the trunk network is also horizontally partitioned and pooled. Though these works seem to introduce a similar idea as our dense supervision, none of them considers decreasing part features’ correlation, which is our main concern.

**Single model ensemble.** Some works which employ

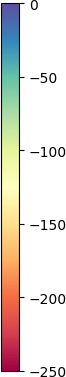
an ensemble of features extracted from a single model are also related to our proposed method. In [29], a hard-aware deeply cascaded network is proposed to employ an ensem- ble of features with different complexities in cascaded man- ner. The cascaded models here are exactly three stages of a GoogleNet[4]. Features in different stages are formed into an ensemble in a boosting manner. [30] proposes to employ an ensemble of features independently in a gradient boost- ing method, which is more similar to our work. However our approach focuses on face recognition problem, and is more similar to bagging method rather than boosting. We use a spatial importance sampling strategy to make different com- ponent features learn from different subsets of the training set, thus reduce correlation between features. Our method is more specific for learning face representation.

In object detection, there are also some related works.

[31] proposes to train Fast R-CNN with adversarially gen- erated occlusion in feature map to simulate occluded sam- ples, where importance sampling is used to sample pixels in feature map to mask out. Another work focuses on face detection[32], which partitions the last feature map into 9 grids and applies multiple supervision on top of them. In spite of the success of part-based models and multi-supervision in many works, there’s no previous work that applies dense supervision in face recognition problem. This motivates us to explore dense supervision’s feasibility to learn better face representation.

## Proposed dense supervision method

In this section, we first introduce how to generate mul- tiple component features with a single face image as input

𝐺(𝑝𝑖)

# (a)

(b)

(c)

**Figure 3:** Visualization of component feature consistency. Less informative region corresponds lower consistency. (a) In most cases training samples are easy to be classified thus component features reach agreement and the variance of consistencies is low.

(b) For profile faces, consistencies vary a lot and show spatial pattern. (c) For faces with severe occlusion, consistencies vary a lot but show no spatial pattern.

(Section3.1). Accordingly, we present the training strat- egy of component features in Section3.2and3.3, which can be further divided into two parts: the definition of the metric *inconsistency* between component features (Section 3.2), and how to use the inconsistency to dynamically se- lect which component features to train in the backward pass (Section3.3). Fig.2demonstrates an overview of the pro- posed approach.

### Component features

We first clarify the difference between single supervision model and our dense supervision model, and then introduce the definition of component features.

**Single supervision.** The most widely used method to train a neural network for face recognition is to apply a single classification supervision on top of the network. The input image goes forward to the stacked convolution layers and

forms a 3D tensor *T* , which is then aggregated to a 2D vector

by global pooling. Usually an embedding layer is added after the global pooling layer to get more compact embedding. Finally the embedding is input to a classifier, which consists of a fully connected layer and a softmax loss.

**Dense supervision.** As illustrated in Fig.2, we modify the single supervision model slightly to apply dense super-

split the output of the backbone network, namely tensor *T* , vision. First, the embedding layer is removed. Second, we into multiple feature vectors. Let’s say the shape of *T* is *c* × *h* × *r*, where *c* is the channel size, *h* is the height and *r* is the width. Then each vector viewed along the channel axis is separated, yielding *h* × *r* feature vectors in total. We

mation of a face, thus each of the *h* × *r* features is input into expect that every feature vector preserves the identity infor-

supervision of all the *h* × *r* softmax losses. an independent classifier. The network is trained under the

In this paper, we define these *h* × *r* features as *compo-*

ResNet[33] with 16× down-sampling rate as our backbone *nent features*. In all the following experiments, we utilize network. The input is resized to 144 × 144, so that the out- put of the backbone network has a shape of 512 × 9 × 9. We

stripe size are both 3 × 3, then the output of the last convo- then add a convolution layer at the end whose kernel size and lution layer is the final tensor *T* with shape 512 × 3 × 3, i.e.

we have nine 512-D component features in total.

### Feature consistency

With the same shared network, we observe that the com- ponent features trained with dense supervision exhibit incon- sistency. A direct reflection of this inconsistency is that if the input image is a hard sample (a profile face or a partial face), the losses of different component features vary a lot (see the second row in Fig.3). We explain this inconsistency as follows: despite all the component features are relatively high-level features and have global receptive fields, the ef- fective receptive field, i.e. the region in the input image that influences the feature most, is limited[34], and prone to be located at the corresponding location. Consequently, the fea- ture whose corresponding effective receptive field is more informative should be more discriminative. An observation

supporting this explanation is that, in our 3 × 3 setting for

component feature split, the component feature at the cen-

ter is always the most discriminative one, i.e. with the low- est loss and highest test accuracy, because the corresponding effective receptive field is centered at the input face’s center. The consistency between component features is of great value. When we employ an ensemble of models for more accurate prediction, the ideal circumstance is a) every single model of the ensemble is accurate and b) every model has low correlation (be inconsistent) with the others, so that they complete each other during test time. In analogy with model ensemble, our dense supervision features can also benefit from low correlation, so we expect component features to have low consistency with each other. To this end, first of all we need to define the consistency between component features. The differences in loss values reflect consistency of component features in some degree. However, simply defining the consistency by the differences between losses involves little information, and when the network is well trained all losses tend to be small thus the differences de- crease rapidly, making it difficult to discover consistency. This motivates us to define the consistency by using a more informative vector, i.e. the class probability vector output by

softmax

***pi*** = ***a***(***Wizi***)*,* (1)

where ***zi*** c R*D* is the *i*th component feature, and ***Wi*** is the *i*th weight matrix of size *C* ×*D*, which turns ***zi*** into per-class response ***Wizi***. Specifically the softmax function is defined as ***a***(***y***) = [*a*1(***y***)*, ..., ai*(***y***)*, ..., aC* (***y***)], where

*C*

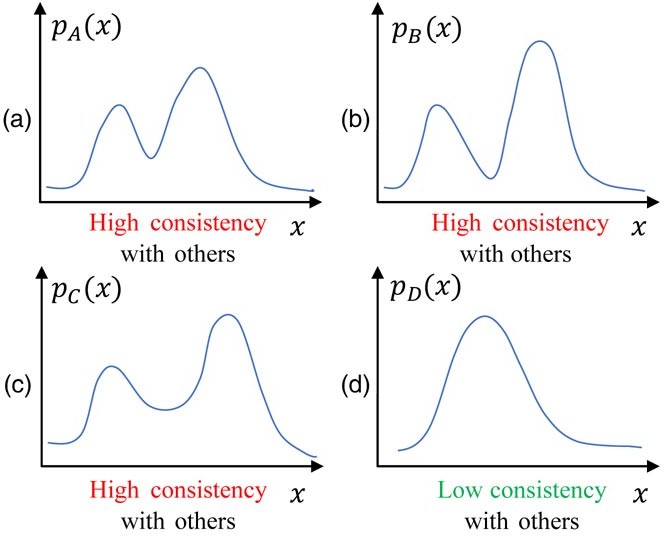
, (2)

*ai*(***y***) = *eyi* / *eyj .*

*j*=1

Let ***pi*** be the softmax probability of the *i*th component feature and ***pj*** be that of the *j*th component feature, Since ***pi*** and ***pj*** are two discrete distributions in the same domain (class 1 to C), we can describe the difference between ***pi*** and ***pj*** by their relative entropy, aka KL divergence. We notate the difference between ***pi*** and ***pj*** as

*C*



**Figure 4:** Illustration on the proposed feature consistency. Fig- ures (a-d) present four different probability distributions, and it can be observed that distributions (a) (b) and (c) are similar to each other, so that the consistency of them with respect to the others is high. In contrast, distribution (d) has low consistency with respect to other distributions.

An intuitive interpretation of the proposed feature con- sistency is illustrated in Fig.4. Suppose we construct a com- plete directed graph, where the softmax probability distri- butions of component features are the vertices and the KL divergence between two probabilities is the weight of the

edge. The graph has to be directed since *dij* Ç *dji*. Then

it is straightforward to notice that, if a vertex has relative

large consistency value, i.e. the sum of its outgoing edges’ weights, it is more likely to be located at the "center" among these vertices. In other words, the component feature with large feature consistency value tends to produce similar clas- sification decision to all the other component features, and vice versa.

### Importance sampling

We visualize component feature consistency of some sam-

ples in the training set for further analysis. An intermediate

*d* = *D*(***p*** ||***p*** ) = , *a* (***W z*** )*log ak*(***Wizi***) *.* (3)

*ij*

***i***

***j***

*k*=1

*k*

***i***

***i***

*ak*(***Wjzj***)

and the visualization is shown in Fig.3. The feature con-

model snapshot is used to generate the feature consistency,

Using a KL divergence metric to evaluate the differences between the pieces of information provided by different com- ponent features, we are not only concentrating on component features’ differences in the probabilities of the ground truth classes, but taking all the classes’ probabilities into consid- eration. This brings more valuable information.

Using the KL divergence as the distance between two softmax probabilities, we define a metric named *feature con- sistency* to evaluate the consistency between one component feature and *all the other* component features. Let it be

,

*Gi* = *G*(*pi*) = − *dij .* (4)

*j*Ç*i*

location in original tensor *T* . sistency is projected to the input image according to their

Some interesting observations can be made from Fig.3. Firstly, the first row shows samples with high average and low variance in feature consistency, which are the major cases in training. The samples have relatively good quality and are easy to be classified, thus with every component feature the classifier performs well and the corresponding softmax dis- tributions are similar to each other.

The second and the third rows show samples with lower average and higher variance in feature consistency. Specif- ically, the second row shows some profile faces. We can

observe that the consistency of each component feature has

Note that *Gi* ≤ 0 always hold.

spatial corresponding relationship with the input image. The

component feature tends to have low consistency with the others if the spatial corresponding region in the input im- age is background. This is natural because if the effective receptive field of the component feature includes too much background information, the feature is prone to be inaccu- rate thus inconsistent with accurate ones. This illustrates that different component features have different effective re- ceptive fields. However, we argue that component features’ effective receptive fields have a large overlap region. We illustrate this argument with examples in the third row of Fig.3, which are some faces with informative regions (eyes) occluded. In such cases, all component feature consistency values increase, rather than only ones that correspond to the occluded regions. Furthermore, the component feature with lowest consistency may not be the one corresponding to the occluded region also. This is because the effective receptive fields of component features have a large overlap so that the occlusion in some region has impact on all the component features. In other words, the component features are cou- pled.

Again we address the two principles to improve the fea- tures’ discriminability: making every component feature dis- criminative and decreasing the correlation between them. To achieve these two goals simultaneously, we utilize the obser- vations above and propose an importance sampling method for training component features based on the feature consis- tency. The main idea is that not all component features are sampled equally for back-propagating. For one specific in-

two groups: the major group S+ and the minor group S−. To put image, we categorize the output component features into

ing order by their consistency. Then the top *k*+ are picked achieve this, we first rank the component features in descend-

minor component features. Component features in S+ are as the major component features and the rest are picked as

considered to be more reliable and more informative, so that

they are assigned large probabilities to be sampled for com-

contrary, component features in S− are considered to be less puting the losses and back-propagating the gradients. On the

informative and less reliable, as they contain different infor-

mation compared with the other component features. One naive strategy is simply to treat minor ones as outliers and ignore them. However, directly ignoring the minor compo- nent features will lead to a trivial solution: The cornered component features are always ignored since they are more likely to be categorized as minor ones. In addition, low con- sistency means the component feature may learn more differ- ent knowledge than the others. So we do not simply ignore the minor component features, but assign them small proba- bilities to sample them for back-propagating. In one forward

*k* |S |

pass, we only sample − from the − minor component

features to back-propagate. Our objective is summarized as

kind of adaptive bagging method. Actually, the component features are not trained with the whole training set uniformly. For each component feature, there exists a subset in the train- ing set where its feature consistency is always high, so that this component feature is always sampled in this subset and tends to learn more from this subset. For the rest of the train- ing set, the consistency tends to be low thus the component feature learns less from it. In this way we decouple the cor- relation between the component features in some degree and increase the diversity of what the component features learn. In practice, we have 9 component features which is di- vided into 2 major ones and 7 minor ones. We sample all the major component features, and 3 of the minor ones, i.e.

*k*+ = 2 and *k*− = 3. We will introduce how *k*+ and *k*− are

selected in Sec.4.2. In testing phase, 9 component features are concatenated and reduced to 1024-D by PCA.

As a summary, the proposed approach consists of two parts. First we propose to train multiple component features within a shared model by applying dense supervision, bring- ing improvements upon the single supervision baseline. Ac- cordingly, we design a balancing strategy to dynamically se- lect which component features to train in a backward pass, further boosting the performance of the dense supervised model.

1. **Experiments**

### Datasets and setting

**Training data.** In our experiments two datasets are used to validate our approach respectively. The first is CASIA- Webface[1], which has 494,414 face images belonging to 10,575 different individuals. It is a widely used training set in face recognition, so we use it to compare our approach with existing works. The second is MS-Celeb-1M[2], which consists of about 100k identities with 10 million images. Due to the memory limitation of the GPU, we select the top 30k identities with the most samples as our large scale training set, leading to around 2.6M samples in total. We name this subset of MS-Celeb-1M MS-30K for convenience. Overlapping identities with test datasets are removed.

**Preprocessing.** We follow the same preprocessing as in SphereFace[9]. MTCNN[35] is used to detect faces and five facial landmarks. All faces are aligned by similarity trans-

from the quotient, i.e. normalize the value to range [−1*,* 1]. formation. Each pixel is divided by 255. Then subtract 1.0

**Backbone network.** For thorough comparison, we use two residual networks[33] with different depth in experiments. One has 20 layers and the other has 64 layers. These two net- works’ structures are almost the same as the one used in [9] except for two differences: first we add batch normalization[36]

after every convolution layer, and second, we add a 3×3 con-

volution layer on top of the original network to reduce the

€ =

,

,−

*ƒi*cS+

€*soƒtmax*

(*ƒi, y*) +

*ƒi*cS

*s*€*soƒtmax*

(*ƒi, y*)*,* (5)

size of output tensor to size 512 × 3 × 3. Thus the only addi- tional computation cost for inference is a 3 × 3 convolution

layer since all BN layers can be integrated into convolution

where *s* = 0*,* 1 indicates whether this component feature is sampled. *ƒi* is the component feature and *y* is the class label.

This importance sampling strategy can be regarded as a

layers.

**Implementation details.** We train all the models fol- lowing the same setting. SGD with momentum 0.9 is used

Ac

k2

k1

k3

curacy 99.10

99.00

98.90

98.80

98.70

98.60

𝑘+ = 3

𝑘+ = 2

𝑘+ = 1

2 3 4 5 𝑘—

perparameters, *k*+ and *k*−. Ideally, *k*+ should not be too much **Figure 5:** Preliminary experiments on the selection of two hy- larger or smaller than *k*−, which results in major/minor com- ponent features dominating the gradient. Also the sum of *k*+ and *k*− should not be too large, which degenerates to the naive dense supervision. We find that *k*+ = 2, *k*− = 3 works well, so *k*+ = 2, *k*− = 3 is used in all our following experiments.

for optimization. The initial learning rate is set to 0.1, and decayed by 0.1 at the 27th, 40th and 48th epoch respec- tively. Training stops at the 54th epoch. Only left-right flip is used for training data augmentation. The batch size is set to 256 for training on CASIA-Webface and 400 for training on MS-30K. We have 9 component features and each is 512-

D. During testing, 9 component features are concatenated and reduced to 1024-D by PCA, then we flip the input image horizontally and extract another set of component features which are also reduced to 1024-D by PCA. Finally the two 1024-D features are element-wisely summed, yielding the final 1024-D face representation.

### Preliminary experiments

selection of two hyperparameters, *k*+ and *k*−. Heuristically, We conduct a group of preliminary experiments on the *k*+ should not be too large or too small and so should *k*− .

using different *k*+ and *k*− and test on LFW[10]. The results We train multiple models with Res20 and CASIA-Webface are shown in Fig.5. As we can observe *k*+ = 2, *k*− = 3 is a

good selection, which is consistent with our suppose that we need to make a trade-off between feature consistency (large

*k*+) and information diversity (large *k*−) while *k*+ and *k*−

should not be too large or too small. Furthermore, in our experiments we find that when the number of component

features varies the optimal ratio of *k*+/*k*− remains almost

constant, i.e. 2/3.

### Performance evaluation

**Evaluation on LFW and YTF.** We firstly evaluate the oposed method on two standard face verification bench- marks. The first one is Labeled Face in the Wild[10] (LFW), which has 13, 233 web-collected images from 5749 differ- ent identities.We following the standard protocol of *unre- stricted with labeled outside data*. 6000 image pairs are pro- vided and split into 10 folds. We report the verification ac- curacy by testing with 10-fold cross validation. The sec- ond benchmark is Youtube Face[14](YTF) dataset, which

pr

includes 3,424 videos from 1,595 different individuals. YTF focuses on video face verification, where a sequence of face images is used to get one face representation rather than a single image. Some samples are more difficult for verifica- tion than still images due to motion blur and low resolution. For simplicity, we average all the features extracted from all frames in a video as the face representation. Similar to LFW, 5000 video pairs are provided and split into 10 folds. We re- port verification accuracy by testing with 10-fold cross vali- dation.

**Table 1**

Face verification accuracy by testing with 10-fold cross validation on LFW[10] and YTF[14], higher is better.

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Training Data | LFW | YTF |
| DeepFace[37] | 4M\* | 97.35 | 91.4 |
| FaceNet[6] | 200M\* | 99.65 | 95.1 |
| Deep FR[38] | 2.6M | 98.95 | 97.3 |
| Baidu[39] | 1.3M\* | 99.13 | N/A |
| Center Loss[7] | 0.7M\* | 99.28 | 94.9 |
| Yi et al. [1] | CASIA | 97.73 | 92.2 |
| Ding and Tao [40] | CASIA | 98.43 | N/A |
| Large Margin[8] | CASIA | 99.10 | 94.0 |
| SphereFace[9] | CASIA | 99.42 | 95.0 |
| 9-ensemble-Res20 | CASIA | 99.27 | 94.7 |
| Softmax-Res20 | CASIA | 98.20 | 92.8 |
| Naive DS-Res20 | CASIA | 98.75 | 94.0 |
| DS-Res20 | CASIA | 99.07 | 94.4 |
| Softmax-Res64 | CASIA | 98.38 | 93.1 |
| Naive DS-Res64 | CASIA | 98.95 | 94.4 |
| DS-Res64 | CASIA | 99.13 | 94.7 |
| Softmax-Res64 | MS-30K | 99.05 | 90.9 |
| Naive DS-Res64 | MS-30K | 99.22 | 93.8 |
| DS-Res64 | MS-30K | 99.50 | 95.4 |

Tab.1reports our experimental results on LFW[10] and TF[14]. We conduct three groups of experiments using different combinations of two training sets and two network architectures. In each group we test 1) a model trained with the single softmax supervision; 2) a model trained using dense supervision without importance sampling, which is denoted as *Naive DS*; and 3) a model trained using dense supervision with importance sampling, which is denoted as *DS*. In the first group we use a 20-layer ResNet[33] as our backbone network and train it with CASIA-WebFace. In the second group, we replace the backbone network in the first group with a 64-layer ResNet[33], while in the last group, we change the training set in the second group to MS-30K. With the same network architecture and training data set, the proposed dense supervision outperforms the corresponding single su-

Y

pervision baseline by a large margin, +0*.*55%,+0*.*57% and

+0*.*17% LFW accuracy in the three groups respectively. Note

that in the last group the baseline’s accuracy is relatively high thus marginal effect makes the improvement smaller. If im- portance sampling is applied to dense supervision, the per- formance is further improved. Compared with state-of-the-

art methods with the same experimental settings (CASIA- Webface + Res64 architecture), our method outperforms the Large Margin Softmax[8], and reaches competitive results with regard to SphereFace[9] (94.7% vs. 95.0% on YTF). Compared previous works trained with more data, our re- sults are also competitive.

We also compare dense supervision with a multi-patch ensemble. The ensemble includes 9 component features. Each one is an independent network and trained separately.

cropped from an 144×144 holistic face around 9 different fa- The input images of the 9 networks are different patches

cial landmarks, such as eye corners, nose tip and mouth cor- ners. Same as in dense supervision, features extracted from 9 networks are concatenated and reduced to 1024-D by PCA. As can be observed, the performance gap between the single model and the multi-patch ensemble is largely closed with

ensemble has 9 times parameters compared with our model. our dense supervision. It is worth noting that the 9-patch

This result shows the superiority of our method compared with the multi-patch ensemble. Though one can carefully search optimal hyperparameters on how to crop patches to get even higher performance, dense supervision only needs training once and the performance is still competitive with much fewer parameters/less computation cost.

**Table 2**

Face verification (Acc@FAR= 0*.*1%) and identification accu- racy (DIR@FAR= 1%) on LFW BLUFR[15], higher is better.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Training | Acc@ | DIR@ | DeepSense V2 | ≤ 0*.*5*M* | 81.29 | 49.37 |
| Data | FAR= 0*.*1% | FAR= 1% | YouTu Lab | ≤ 0*.*5*M* | 83.29 | 61.61 |

Method

than standard LFW. Some methods achieve around 99% on ported1. The LFW BLUFR benchmark is more challenging

standard LFW, but the verification accuracy and DIR at low false accept rate remain quite low.

Tab.2reports our experimental results on LFW BLUFR benchmark. Our baseline (Softmax-Res64) performs well

in standard LFW with over 98% verification accuracy, how-

turns to be not good enough. For identification, the DIR@FAR= ever, when tested on BLUFR benchmark, the performance 1% is only 63*.*23%. Dense supervision (DS-Res64) signifi-

cation rate at low false accept rate, from 89*.*76% to 98*.*91% cantly improves the verification accuracy and the identifi- and 63*.*23% to 91*.*85% respectively. This large gap shows

the superiority of our method especially on large scale face retrieval problem in low FAR scenario. Our method out- performs LightCNN[43] which has state-of-the-art perfor- mance on BLUFR benchmark. Note that LightCNN[43] is trained with more data, i.e. the whole MS-Celeb-1M dataset, while ours only use a subset.

**Table 3**

Top-1 face identification accuracy on MegaFace[11], with one million distractors. Higher is better.

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Data | FaceScrub | FGNet |
| Beijing FaceAll Norm 1600 | ≤ 0*.*5*M* | 64.80 | 25.02 |
| Google - FaceNet v8 | ≤ 0*.*5*M* | 70.49 | 74.59 |
| NTechLAB - facenx large | ≤ 0*.*5*M* | 73.30 | 52.72 |
| SIATMMLAB TencentVision | ≤ 0*.*5*M* | 74.20 | 71.25 |

C-contrastive[42] CASIA 95.83 77.18

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HighDimLBP [41] | N/A | 41.66 | 18.07 | Vocord - deepVo V3 | ≤ 0*.*5*M* | 91.76 | 61.40 |
| Center Loss[7] | 0.7M | 93.35 | 67.86 | SphereFace[9] | CASIA | 72.73 | N/A |
| Yi et al. [1] | CASIA | 80.26 | 28.90 | SphereFace(3patch)[9] | CASIA | 75.77 | 47.56 |
| L2Norm[42] | CASIA | 95.77 | 73.92 | CosFace[44] | 5M | 82.76 | N/A |
| C-triplet + center[42] | CASIA | 95.73 | 76.12 | CosFace(3patch)[44] | 5M | 84.26 | 67.87 |

LightCNN[43] MS-1M 98.71 90.42

Softmax-Res64 MS-30K 89.76 63.23

DS-Res64 MS-30K 98.91 91.85

**Evaluation on LFW BLUFR.** The standard LFW pro- tocol is very limited, with only 3,000 positive and 3,000 neg-

ative pairs for verification. Accuracy over 98% can be eas-

ily achieved on LFW, leaving little room for algorithm de- velopment. Thus we also use another benchmark, the LFW Benchmark of Large-scale Unconstrained Face Recognition (BLUFR) [15], for algorithm evaluation. LFW BLUFR has two protocols: face verification and face identification. For testing with face verification protocol, there are 10-fold ex- periments, with each fold containing about 156,915 positive pairs and 46,960,863 negative pairs on average, and the veri-

fication accuracy at 0*.*1% false accept rate (Acc@FAR=0*.*1%)

is reported. For testing with face identification protocol, the dataset is split into a genuine probe set, an impostor probe

set and a gallery set. The experiment is repeated for 10 times

and the average detection and identification rate (DIR) is re-

|  |  |  |  |
| --- | --- | --- | --- |
| Softmax-Res64 | MS-30K | 70.34 | 27.31 |
| DS-Res64 | MS-30K | 80.81 | 47.01 |

**Evaluation on MegaFace.** MegaFace[11] is a challeng- ing benchmark up to date for both face retrieval and face ver- ification, which consists of two probe sets and a gallery set. The probe sets are two existing datasets: FaceScrub[46] and FGNet. FaceScrub[46] dataset contains 106,863 face images of 530 celebrities, while FGNet is a relatively small dataset, mainly focusing on cross-age face recognition. It consists of 1002 face images of 82 subjects at different ages. The gallery set is constructed by mixing up target images (face images selected from the same subjects as those in probe sets) and a set of one million distractors, which have no overlap with the probe subjects. For testing with the face retrieval pro- tocol, K-nearest neighbors of the probe images are found in the gallery set and the Rank-1 accuracy is reported. Due to the huge amount of distractors, MegaFace is one of the most challenging benchmarks for face recognition. We re- port Rank-1 retrieval accuracy with both FaceScrub[46] and

1We refer readers to [15] for definition of the DIR.

**Table 4**

Cross-view face verification similarities on VGGFace2. Higher is better.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Data | 0○ − 0○ | 0○ − 45○ | 0○ − 90○ | 45○ − 0○ | 45○ − 45○ | 45○ − 90○ | 90○ − 0○ | 90○ − 45○ | 90○ − 90○ |
| Softmax[3] | VGGFace2 | 0.6876 | 0.6821 | 0.6222 | 0.6859 | 0.6980 | 0.6481 | 0.6264 | 0.6515 | 0.6488 |
| Softmax[45] | VGGFace2 | 0.8397 | 0.8262 | 0.7325 | 0.8319 | 0.8486 | 0.7672 | 0.7386 | 0.7704 | 0.7805 |
| Softmax[3] | MS-1M | 0.5661 | 0.5582 | 0.4715 | 0.5628 | 0.5766 | 0.5036 | 0.4776 | 0.5064 | 0.5094 |
| Softmax[45] | MS-1M | 0.8605 | 0.8427 | 0.6693 | 0.8486 | 0.8572 | 0.6993 | 0.6718 | 0.7050 | 0.7172 |
| Softmax[45] | CASIA | 0.8158 | 0.7878 | 0.6483 | 0.7946 | 0.8135 | 0.6955 | 0.6515 | 0.7015 | 0.7233 |
| Center Loss[45] | CASIA | 0.8128 | 0.8001 | 0.7018 | 0.8088 | 0.8199 | 0.7328 | 0.7081 | 0.7332 | 0.7331 |
| SphereFace[45] | CASIA | 0.8010 | 0.7678 | 0.6211 | 0.7784 | 0.7934 | 0.6708 | 0.6262 | 0.6749 | 0.6957 |
| Softmax-augUV[45] | CASIA | 0.8342 | 0.8182 | 0.7237 | 0.8256 | 0.8404 | 0.7582 | 0.7302 | 0.7597 | 0.7682 |
| Softmax | MS-30K | 0.7832 | 0.7507 | 0.6027 | 0.7601 | 0.7849 | 0.6596 | 0.6022 | 0.6603 | 0.7259 |
| DS | MS-30K | 0.8725 | 0.8563 | 0.7533 | 0.8591 | 0.8712 | 0.7780 | 0.7541 | 0.7804 | 0.8131 |

FGNet as probe sets.

Our baseline (Softmax-Res64) performs well (70*.*34% Rank- Experimental results on MegaFace are shown in Tab.3.

1 accuracy) when using FaceScrub as the probe set, but has a poor performance when using FGNet as the probe set. This is because our training data do not have large variation on ages. The proposed dense supervision (DS) outperforms the

baseline by a large margin, with both FaceScrub[46] (+10*.*47%)

and FGNet (+19*.*70%) as probe sets. Compared with pub-

lished state-of-the-art results, our method is competitive in FaceScrub with regard to CosFace[44] which uses a more discriminative loss function and a high quality private dataset for training. Compared with unpublished commercial al- gorithms which trained with large scale private data, our method is also competitive.

**Evaluation on VGGFace2.** VGGFace2[3] contains 3.31 million images of 9131 subjects, with an average of 362.6 images for each subject. This dataset has much larger vari- ations on faces’ pose, age, illumination, ethnicity and pro- fession than other existing datasets such as CASIA and MS- Celeb-1M. The dataset is split into a training set and a test set. In this study, we only use the test set for algorithm eval- uation. The test set contains 500 subjects, and includes two protocol, i.e. face matching across different poses, and face matching across different ages. This experiment aims at as- sessing the performance of our proposed method when faces have large variation in pose, so we follow the pose matching protocol.

In the pose matching protocol, the unit of face represen- tation is the *pose template*, which comprises five faces of each subject in a same pose. We simply average the five faces’ features to get a template feature. Pose templates are

sorted into three types, i.e. frontal (0○), three-quarter(45○),

and profile (90○). A subset of the test set which contains 300

subjects is used for pose matching protocol, and each sub- ject provides 2 templates for each pose view. Consequently there are 6 templates per subject and 1.8K templates in total. These six templates are divided into two sets, such that each one contains templates for all the three poses. Then the pair- wise cosine similarity between templates in the two sets is

computed, yielding a matrix *Di* c R3×3. Finally the average

.*D* = *Di*

across all subjects 1 300 is reported as the metric of pose matching perfor3m00ance*i*=. 1

Tab.4showcases our results on VGGFace2 pose match- ing protocol. We first compare baselines in two existing works[3,45]. Both of the two provide baselines with VG- GFace2 and MS-1M as training sets and softmax as loss function. However, even with a shallower network archi- tecture (ResNet-27 vs. ResNet-50), baselines in [45] out- perform those in [3] by a large margin. We are inclined to believe that results in [45] are more reasonable, because our baseline, which is trained with ResNet64 + softmax + MS- 30K, has more similar results to those reported in [45]. One should also notice that, when preprocessing the test data, in

around 10% images MTCNN fails to detect face due to the

large variation in pose. So for the failed images, we roughly crop the faces and resize them to the size of aligned faces,

which means there are around 10% inaccurately-aligned faces

in our test set. This explains why our softmax baseline trained with MS-30K performs even worse than that trained with CASIA in [45]. However, even with inaccurately-aligned

(Softmax-augUV) in all the 3 × 3 cross-pose similarities. faces, our proposed method outperforms state-of-the-art[45]

Note that [45] makes great efforts on pose augmentation to learn pose invariant face representation, while our method concentrates on learning rich part-sensitive features and is more concise.

The proposed method also outperforms the correspond- ing single supervision baseline by a large margin. Specially, we observe that the improvement in profile-frontal face match-

ing is the most significant(see 0○−90○ and 90○−0○ in Tab.4).

This improvement shows the robustness of our method against pose variation.

## Conclusion

In this paper we propose a novel face recognition method based on the dense supervision strategy. Unlike most previ- ous works which aim at designing a single discriminative loss function, we apply multiple classification loss on top of multiple component features extracted from a single net- work. To make component features more accurate and less correlated with each other, we first proposes a metric called feature consistency to evaluate the correlation between one component feature and the others, then use it to sample appli- cable component features for learning. Experimental results show that our proposed method is effective when compared

with several existing representative methods.

## Conflict of interest statement

We declare that we have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

## Acknowledgment

This work is supported by the Basic Special Project of Ministry of Public Security of China under grant no. 2016GA BJC01 and the National Key R&D Program of China under grant no. 2016YFC0801003.

## References

1. Yi, D., Lei, Z., Liao, S., Li, S.Z., 2014. Learning face representation from scratch. arXiv preprint arXiv:1411.7923 .
2. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J., 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition, in: In Proc. ECCV Conf., pp. 87–102.
3. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A., 2018. Vg- gface2: A dataset for recognising faces across pose and age, in: In Proc. FG Conf., pp. 67–74.
4. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: In Proc. CVPR Conf., pp. 1–9.
5. Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, in: In Proc. CVPR Conf., pp. 7132–7141.
6. Schroff, F., Kalenichenko, D., Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering, in: In Proc. CVPR Conf., pp. 815–823.
7. Wen, Y., Zhang, K., Li, Z., Qiao, Y., 2016. A discriminative feature learning approach for deep face recognition, in: In Proc. ECCV Conf., pp. 499–515.
8. Liu, W., Wen, Y., Yu, Z., Yang, M., 2016. Large-margin softmax loss for convolutional neural networks, in: In Proc. ICML Conf., pp. 507–516.
9. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L., 2017. Sphereface: Deep hypersphere embedding for face recognition, in: In Proc. CVPR Conf., pp. 6738–6746.
10. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E., 2007. La- beled faces in the wild: a database for studying face recognition in unconstrained environments. Tech. rep. .
11. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E., 2016. The megaface benchmark: 1 million faces for recognition at scale, in: In Proc. CVPR Conf., pp. 4873–4882.
12. Deng, J., Guo, J., Xue, N., Zafeiriou, S., 2019. Arcface: Additive an- gular margin loss for deep face recognition, in: In Proc. CVPR Conf., pp. 4690–4699.
13. Sun, Y., Wang, X., Tang, X., 2014. Deep learning face representation from predicting 10, 000 classes, in: In Proc. CVPR Conf., pp. 1891– 1898.
14. Wolf, L., Hassner, T., Maoz, I., 2011. Face recognition in uncon- strained videos with matched background similarity, in: In Proc. CVPR Conf., pp. 529–534.
15. Liao, S., Lei, Z., Yi, D., Li, S.Z., 2014. A benchmark study of large- scale unconstrained face recognition, in: In Proc. IJCB Conf., pp. 1–8.
16. Wen, G., Mao, Y., Cai, D., He, X., 2018. Split-net: Improving face recognition in one forwarding operation. Neurocomputing 314, 94– 100.
17. Oh, B.S., Oh, K., Teoh, A.B.J., Lin, Z., To, K.A., 2017. A gabor-

based network for heterogeneous face recognition. Neurocomputing 261, 253–265.

1. Wang, P., Su, F., Zhao, Z., Guo, Y., Zhao, Y., Zhuang, B., 2019. Deep class-skewed learning for face recognition. Neurocomputing 363, 0925–2312.
2. Sun, Y., Chen, Y., Wang, X., Tang, X., 2014. Deep learning face representation by joint identification-verification, in: In Proc. NIPS Conf., pp. 1988–1996.
3. Rashedi, E., Barati, E., Nokleby, M., wen Chen, X., 2019. Stream lossâǍI: Convnet learning for face verification using unlabeled videos in the wild. Neurocomputing 329, 311–319.
4. Hadsell, R., Chopra, S., LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping, in: In Proc. CVPR Conf., pp. 1735– 1742.
5. Ding, C., Tao, D., 2018. Trunk-branch ensemble convolutional neural networks for video-based face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 40, 1002–1014.
6. Yao, H., Zhang, S., Hong, R., Zhang, Y., Xu, C., Tian, Q., 2019. Deep representation learning with part loss for person re-identification. IEEE Trans. Image Process. 28, 2860–2871.
7. Huang, Y., Sheng, H., Zheng, Y., Xiong, Z., 2017. Deepdiff: Learn- ing deep difference features on human body parts for person re- identification. Neurocomputing 241, 191–203.
8. Sun, Y., Zheng, L., Yang, Y., Tian, Q., Wang, S., 2018. Beyond part models: Person retrieval with refined part pooling (and A strong con- volutional baseline), in: In Proc. ECCV Conf., pp. 501–518.
9. Zhong, W., Jiang, L., Zhang, T., Ji, J., Xiong, H., 2019. Combin- ing multilevel feature extraction and multi-loss learning for person re-identification. Neurocomputing 334, 68–78.
10. Zhao, H., Tian, M., Sun, S., Shao, J., Yan, J., Yi, S., Wang, X., Tang, X., 2017. Spindle net: Person re-identification with human body re- gion guided feature decomposition and fusion, in: In Proc. CVPR Conf., pp. 907–915.
11. Zhang, X., Luo, H., Fan, X., Xiang, W., Sun, Y., Xiao, Q., Jiang, W., Zhang, C., Sun, J., 2017. Alignedreid: Surpassing human- level performance in person re-identification. arXiv preprint arXiv: 1711.08184 .
12. Yuan, Y., Yang, K., Zhang, C., 2017. Hard-aware deeply cascaded embedding, in: In Proc. ICCV Conf., pp. 814–823.
13. Opitz, M., Waltner, G., Possegger, H., Bischof, H., 2017. BIER - boosting independent embeddings robustly, in: In Proc. ICCV Conf., pp. 5199–5208.
14. Wang, X., Shrivastava, A., Gupta, A., 2017. A-fast-rcnn: Hard posi- tive generation via adversary for object detection, in: In Proc. CVPR Conf., pp. 3039–3048.
15. Opitz, M., Waltner, G., Poier, G., Possegger, H., Bischof, H., 2016. Grid loss: Detecting occluded faces, in: In Proc. ECCV Conf., pp. 386–402.
16. He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: In Proc. CVPR Conf., pp. 770–778.
17. Luo, W., Li, Y., Urtasun, R., Zemel, R.S., 2016. Understanding the effective receptive field in deep convolutional neural networks, in: In Proc. NIPS Conf., pp. 4898–4906.
18. Zhang, K., Zhang, Z., Li, Z., Qiao, Y., 2016. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Proc. Let. 23, 1499–1503.
19. Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: In Proc. ICML Conf., pp. 448–456.
20. Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. Deepface: Clos- ing the gap to human-level performance in face verification, in: In Proc. CVPR Conf., pp. 1701–1708.
21. Parkhi, O.M., Vedaldi, A., Zisserman, A., 2015. Deep face recogni- tion, in: In Proc. BMVC Conf., pp. 1–12.
22. Liu, J., Deng, Y., Bai, T., Huang, C., 2015. Targeting ultimate accuracy: Face recognition via deep embedding. arXiv preprint arXiv:1506.07310 .
23. Ding, C., Tao, D., 2015. Robust face recognition via multimodal deep face representation. IEEE Trans. Multimedia 17, 2049–2058. [41]Ahonen, T., Hadid, A., Pietikainen, M., 2006. Face description with

local binary patterns: Application to face recognition. IEEE Trans. Pattern Anal. 28, 2037–2041.

2

1. Wang, F., Xiang, X., Cheng, J., Yuille, A.L., 2017. Normface: L hypersphere embedding for face verification, in: In Proc. ACMMM Conf., pp. 1041–1049.
2. Wu, X., He, R., Sun, Z., 2015. A lightened CNN for deep face repre- sentation. arXiv preprint arXiv:1511.02683 .
3. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W., 2018. Cosface: Large margin cosine loss for deep face recogni- tion, in: In Proc. CVPR Conf., pp. 5265–5274.
4. Deng, J., Cheng, S., Xue, N., Zhou, Y., Zafeiriou, S., 2018. UV- GAN: adversarial facial UV map completion for pose-invarian face recognition, in: In Proc. CVPR Conf., pp. 7093–7102.
5. Ng, H., Winkler, S., 2014. A data-driven approach to cleaning large face datasets, in: In Proc. ICIP Conf., pp. 343–347.

Zhiqiang Liu received the B.S. degree in material forming and control engineering from Jiaozuo In- stitute of Technology, Jiaozuo, China, in 2003 and the M.S. degree in management science and en- gineering from Hebei University of Engineering, Handan, China, in 2006. His research interests in- clude the security system and engineering, and im- age processing.

Weijun Hong is a professor at People’s Public Se- curity University of China, and he is a special ex- pert in video surveillance of public security of the Ministry of Public Security of China. His research interests include security system and engineering, and image processing.

Hongzhou Zhang is an associate professor at Peo- ple’s Public Security University of China. His re- search interests include security system and engi- neering, and image processing.

Jianhui Ma is a Commissioner General Class 2 of Police Technology and Police Commissioner Class 3 in the Bureau of Science and Technology Infor- mation of the Ministry of Public Security of China. His research interests include security system and engineering, and big data.

## Author Contribution

Zhiqiang Liu: Methodology, Writing-Original draft preparation, Writing- Reviewing and Editing, Software, Visualization, Validation. Weijun Hong: Conceptualization, Supervision, Funding acquisition, Investigation. Hongzhou Zhang: Conceptualization, Investigation, Supervision, Formal analysis. Jian- hui Ma: Data curation, Resources, Investigation, Validation.

